Institute of Acoustics

AUDIO SPECTRAL ANALYSIS USING THE WINOGRAD FOURIER TRANSFORM

J A S Angus (1) and M T Whitaker (2)

(1) Electronics Department, University of York, Heslington, York
(2) Data Conversion Systems, Cambridge

INTRODUCTION:

Spectral analysis is the basis of many audio measurement systems, e.g. noise meters, TDS systems, etc. Currently this analysis may be done in real-time via special purpose analogue or digital hardware or in non real-time via software. In many cases a real-time (or near real-time) analysis is necessary and so the instrument designer is forced to grapple with the problems and additional cost of special purpose hardware. It would be desirable to be able to perform real-time spectral analysis using standard microprocessor components as these offer both cost and development advantages.

The Winograd Fourier Transform (WFT) is a discrete Fourier transform (DFT) algorithm which minimises the number of multiplications required to perform the transform by using a more complex sequence of calculations. However, the reduction in the number of multiplications makes the algorithm suitable for implementation in real-time on a standard microprocessor. The rest of this paper describes the WFT algorithm and its implementation.

THEORETICAL BACKGROUND

The Winograd Fourier Transform is an algorithm that can be used to calculate the DFT when the transform length can be factored into a set of small prime numbers (or small prime powers). All the factors must be mutually prime (that is, each possible pair of factors have an h.c.f of 1), so that repeated factors or more than one power of the same prime are not allowed. The algorithm works by first converting the one-dimensional array of data into a multi-dimensional array via a simple one to one mapping of the data. The Fourier transform of the resulting multi-dimensional array can be calculated by carrying out a Fourier transform along each dimension in turn, which is performed by using some extremely efficient small length DFT algorithms developed by Winograd [1].

The resulting transformed data is then converted back into a one-dimensional array via a second mapping. To illustrate the operations being carried out, the data flow for a 15 point transform is shown in figure 1. This transform is performed by a two-dimensional transform with factors of 3 and 5.

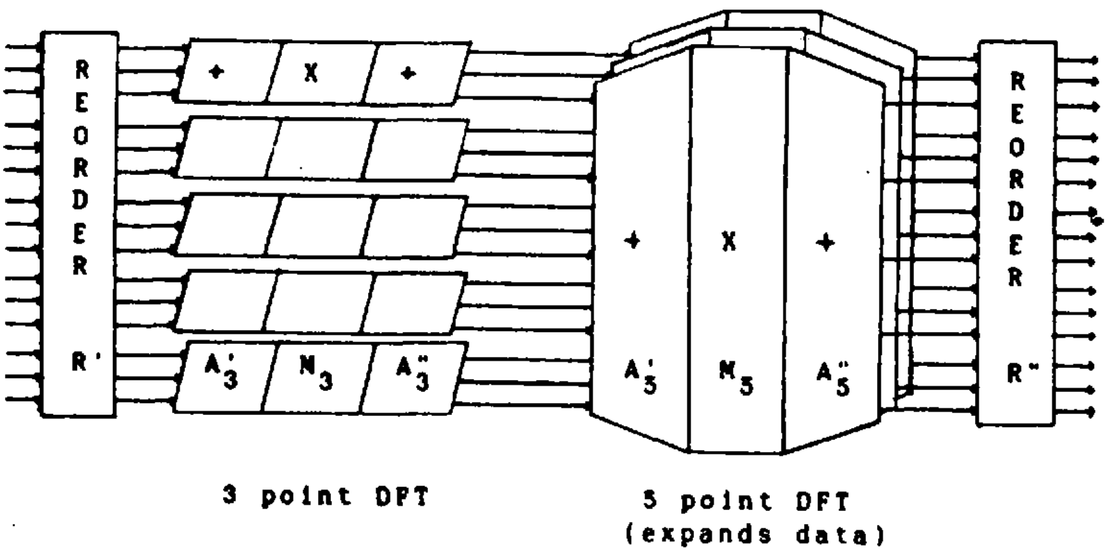

Figure 1 Data Flow for 15 Point WFT

The one to many dimension transform used was first described by Good [2]. He gives two one to one mappings between the one-dimensional space and the multi-dimensional space, these being called the Sino correspondence and the Ruritanian correspondence. For a one-dimensional space of dimension N and a multi-dimensional space of dimensions $(N_1,N_2..N_n)$ (where $N = N_1 \times N_2 \times ..N_n$) and defining

$$R_i = \frac{N}{N_i} \qquad \text{and} \qquad S_i = k_i R_i \qquad - 1$$

(where k_i is the least value such that $(k_i R_i \bmod N_i) = 1$)
the Sino correspondence between points $[x]$ and $[x_1,x_2..x_n]$ is given by

$$x = (S_1 x_1 + S_2 x_2 + ..S_n x_n) \bmod N \qquad - 2$$

and the Ruritanian correspondence is given by

$$x = (R_1 x_1 + R_2 x_2 + ..R_n x_n) \bmod N \qquad - 3$$

These correspondances are based on the Chinese Remainder Theorem [3] and so rely on the dimensions $(N_1,N_2..N_n)$ being mutually prime, which gives one of the constraints on transform size.

Good then shows that if the Sino correspondance is applied to the input of a multi-dimensional DFT and the Ruritanian correspondence is applied to the output, then the result is equivalent to the result of a one-dimensional DFT. It should be noted that the proof is still valid if the Ruritanian correspondence is used at the input of the WFT and the Sino correspondance is used at the output.

The small length DFT algorithms used to calculate the multi-dimensional transform were derived by Winograd [1] by representing the DFTs in matrix form, and reordering the matrices to turn them into cyclic convolutions, using the Rader prime algorithm [4]. The idea behind this algorithm is that the order of evaluation within the summation term of the DFT (see equation 5) is unimportant, as is the order in which the frequency terms are calculated, thus any permutation of the indices i and k that makes the calculation easier to perform may be applied. This permutation corresponds to interchanging rows or columns in the matrix representations of the transform. Determining a permutation that puts the matrix in the form of a cyclic convolution allows the application of fast convolution algorithms, such as those developed by Winograd.

The Radar algorithm relies on the size of the small DFT, N, being a prime number, which means a number, r, can be found such that there is a one to one mapping between the integers n and the integers r(n) (where both n and r(n) lie in the range 1..N-1), given by

$$r(n) = (r^n) \bmod N \qquad - 4$$

The short-time DFT, ignoring the windowing operation, is given by

$$X[k] = \sum_{i=0}^{N-1} x[i]w[ik] \quad \text{where} \quad w[ik] = e^{-j2\pi ik/N} \qquad - 5$$

Treating x[0] and X[0] as special cases gives the following pair of equations

$$X[0] = \sum_{i-0}^{N-1} x[i] \quad \text{and} \quad X[k] - X[0] = \sum_{i=1}^{N-1} x[i](w[ik] - 1) \qquad - 6$$

Applying the permutation of equation 4 to equation 6, so that $i \to r(i)$ and $k \to r(k)$ gives

$$X[r(k)] - X[0] = \sum_{i=1}^{N-1} x[r(i)](w[r(i)r(k)] - 1) \qquad - 7$$

and as w[n] = w[n mod N]

$$X[r(k)] - X[0] = \sum_{i=1}^{N-1} x[r(i)](w[r(i+k)] - 1) \qquad - 8$$

Given that for integers, n, outside the range 1..N, r(n) = r(n mod N-1) from Euler's theorem) and as r(n) lies within the range 1..n-1 (by definition), equation 8 can be recognised as a cyclic correlation, and, by reversing the order of the sequence x[r(i)], can be transformed into a cyclic convolution.

Winograd shows that the Rader prime algorithm can be extended to transform lengths that are powers of a prime. Then, using a technique developed for calculating cyclic convolutions (designed to minimise the number of multiplications required) he decomposes each reordered matrix into a series of additions (which may expand the size of the input data), a series of multiplications, and a second series of additions (which will contract the data back down to the original size). These may be represented in matrix form by

$$W = A'' \; M \; A' \qquad - 9$$

where W is the small DFT operation, A' is the first series of additions (hereafter referred to as the pre-weave matrix), M is the series of multiplications, and A" is the second series of additions (the post-weave matrix). A' and A" consist solely of ones and zeroes, while M is a diagonal matrix whose coefficients are either purely real or purely imaginary. This means that only two real multiplications are required for each complex multiplication, rather than the four that would be required if the coefficients were complex. Winograd includes algorithms for 2, 3, 4, 5, 7, 8, 9 and 16 point DFTs in his article [1] - larger sizes than this are relatively inefficient. As an example of the operations involved, the derivation of the 5 point DFT algorithm is given in an Appendix at the end of this paper.

A further reduction in the number of multiplications required by the complete algorithm is achieved by reordering the calculation of the small DFTs. Consider the matrix form of figure 1.

$$W_{15} = R'' \; W_5 \; W_3 \; R' = R'' \; A''_5 \; M_5 \; A'_5 \; A''_3 \; M_3 \; A'_3 \; R' \qquad - 10$$

where R' is the input reordering matrix and R" is the output reordering matrix.

Due to the special characteristics of the addition and multiplication matrices, this equation can be reordered as follows

$$W_{15} = R'' \; A''_3 \; A''_5 \; M_5 \; M_3 \; A'_5 \; A'_3 \; R' \qquad - 11$$

and as M_3 and M_5 are diagonal matrices, multiplying them together will give one large diagonal matrix containing all the multiplications needed by the transform. The validity of this reordering may be proved either by matrix algebra [5] or by conventional algebra [6]. The data flow for the reordered equation is illustrated in figure 2.

AUDIO SPECTRAL ANALYSIS USING THE WINOGRAD FOURIER TRANSFORM

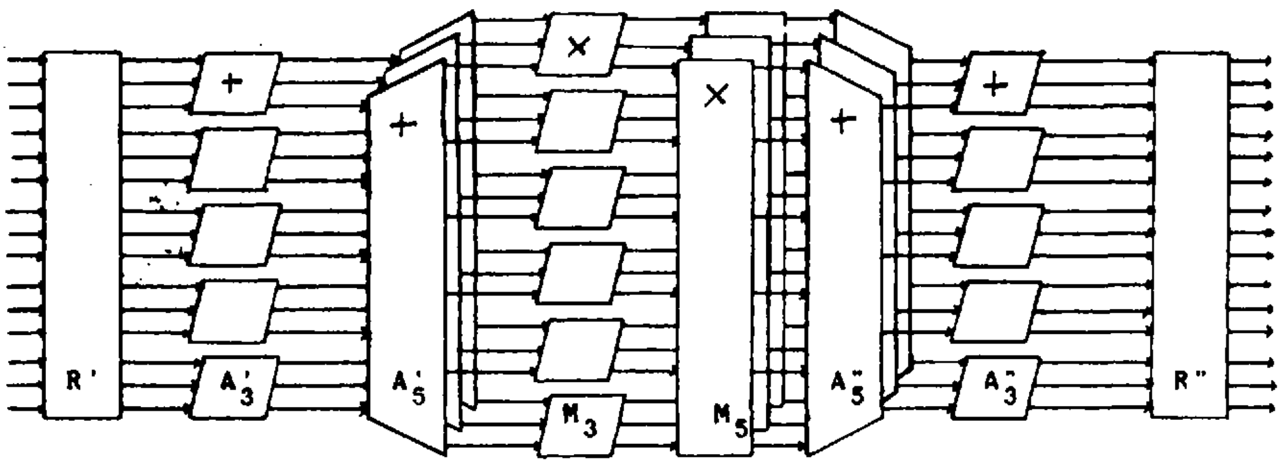

Figure 2 Data Flow in Reordered 15 Point WFT

Several advantages are gained by using the WFT rather than the FFT apart from the obvious one of speed. Having only one stage of multiplication makes scaling the transform easier, and reduces rounding errors. Also, as real and imaginary data are separate until after the multiplication stage, real transforms take approximately half the time of full complex ones. The disadvantage is algorithmic complexity - the FFT may be coded in a few lines of a high level language, while the WFT takes several hundred lines.

HARDWARE IMPLEMENTATION

The hardware required for a speech analysis system based on the WFT includes an A/D convertor, a microprocessor, some memory and some means of transmitting the resulting frequency information. For the A/D convertor the Reticon R5640 was used, which incorporates an anti-aliasing filter, sample and hold circuit and a 10 bit A/D convertor on one chip. The processor used was the Motorola 68010 (running at 10MHz), selected for its ease of programming, large variety of addressing modes, and speed. The 68010 version was selected because it performs (multiplications) significantly faster than the standard 68000. 16K bytes of CMOS static RAM and 16K bytes of CMOS EPROM were used for memory, as these have fast enough access speeds to allow the 68010 to run at 10MHz with no wait states. For transmission of data a 68230 parallel interface was used. The total cost of this system was under £100.

SOFTWARE IMPLEMENTATION

For high quality speech analysis a frequency bandwidth of 6kHz is needed, requiring imput sampling at 12kHz. A frequency resolution of 100Hz, with a new transform every 10msec is adequate to resolve transitory information. A 120 point transform, performed in under 10 msec is therefore required. This factors into a (3 x 5 x 8) transform which is one of the more efficient sizes to use as both the 3 point and 8 point DFT algorithms cause no expansion of the input data.

Separate pre-weave and post-weave routines were written for each of the small DFTs. Because the data must be reordered between each routine (the input reordering R', intermediate dimension reordering, and the output reordering R") the pre-weave routines are designed to access their input data via index arrays, while the post-weave routines do the same with their output data. These index arrays can be pre-calculated, along with the array of multipliers. This system of data access can be implemented very efficiently on the 68000, due to the existence of a post-increment register indirect addressing mode, and is more efficient than the normal means of data access using nested loops. It also means that no separate reordering routine is required for the input and output reordering, as all pre-weave and post-weave routines have it built in. A sample routine for the 5 point DFT, and the outer routine for the WFT, both written in Modula-2 are given in appendix 1. For the final code used by the 68000 all routines were written in assembly language, the multiplication routine was combined with the output of the 5 point pre-weave stage, and the parts of the pre-weave and multiplication routines that deal with the imaginary half of the data were deleted. Other tasks performed are data sampling, windowing (using the Hanning window), log magnitude calculation and data output. 16 bit integer arithmetic is used throughout the program, which means that the only scaling required is truncation back to 16 bits after the multiplication stage. The constants in the multiplication stage must, however, be scaled so that the largest constant equals 32767, to give the greatest possible numerical accuracy.

PERFORMANCE

The following table compares the time taken by the WFT and FFT for two sizes of transform and several different implementations. Timings for the 32010 are from reference [7]. All timings are for a full complex transform.

Processor	Language	120pt WFT	128pt FFT	240pt WFT	256pt FFT
8088 @4.77MHz	Modula-2	170ms	290ms	370ms	660ms
	C	85ms	190ms	190ms	440ms
68010 @ 10MHz	C	18ms	50ms	42ms	110ms
	Assembly	9ms	-	-	-
32010 @ 20MHz	Assembly	-	7ms	-	16ms

A program has been written to display the results obtained from the transform board on the IBM PC. Time is displayed horizontally, frequency vertically, and amplitude by digital halftoning using a 4 by 4 cell [8]. A sample spectrogram is shown in figure 3.

CONCLUSION

This paper has described the WFT and its use as a technique for providing real time spectral analysis on a standard microprocessor. The broader application of this technique has the potential for providing lower cost audio measurement systems.

REFERENCES

[1] S. WINOGRAD (1978) "On Computing the Discrete Fourier Transform". Mathematics of Computation, Vol. 32. No. 141 p 175.

[2] I.J. GOOD (1971) "The Relationship Between Two Fast Fourier Transforms". IEE Trans. Computers, Vol. 20, No. 3, p 310.

[3] J.H.McCLELLAN, C.M. RADER (1979) "Number Theory in Digital Signal Processing". Prentice Hall, Englewood Cliffs, N.J.Chapter 7, p 56.

[4] C.M. RADER (1968) "Discrete Fourier Transforms When the Number of Data Samples is Prime". Proc. IEEE, Vol. 56, No. 6, p 1107.

[5] R.E. BLAHUT (1984) "Fast Algorithms for Digital Signal Processing". Addison-Wesley, Reading, MA. Chapter 8, p 249.

[6] D.P. KOLBA, T.W. PARKES (1977) "A Prime Factor FFT Algorithm Using High Speed Convolution". IEEE Trans. Acoustics, Speech and Signal Processing, Vol. 25, No. 4, p 281.

[7] S. MAGER, R. HESTER, R. SIMPSON (1982) "Signal-processing μC Builds

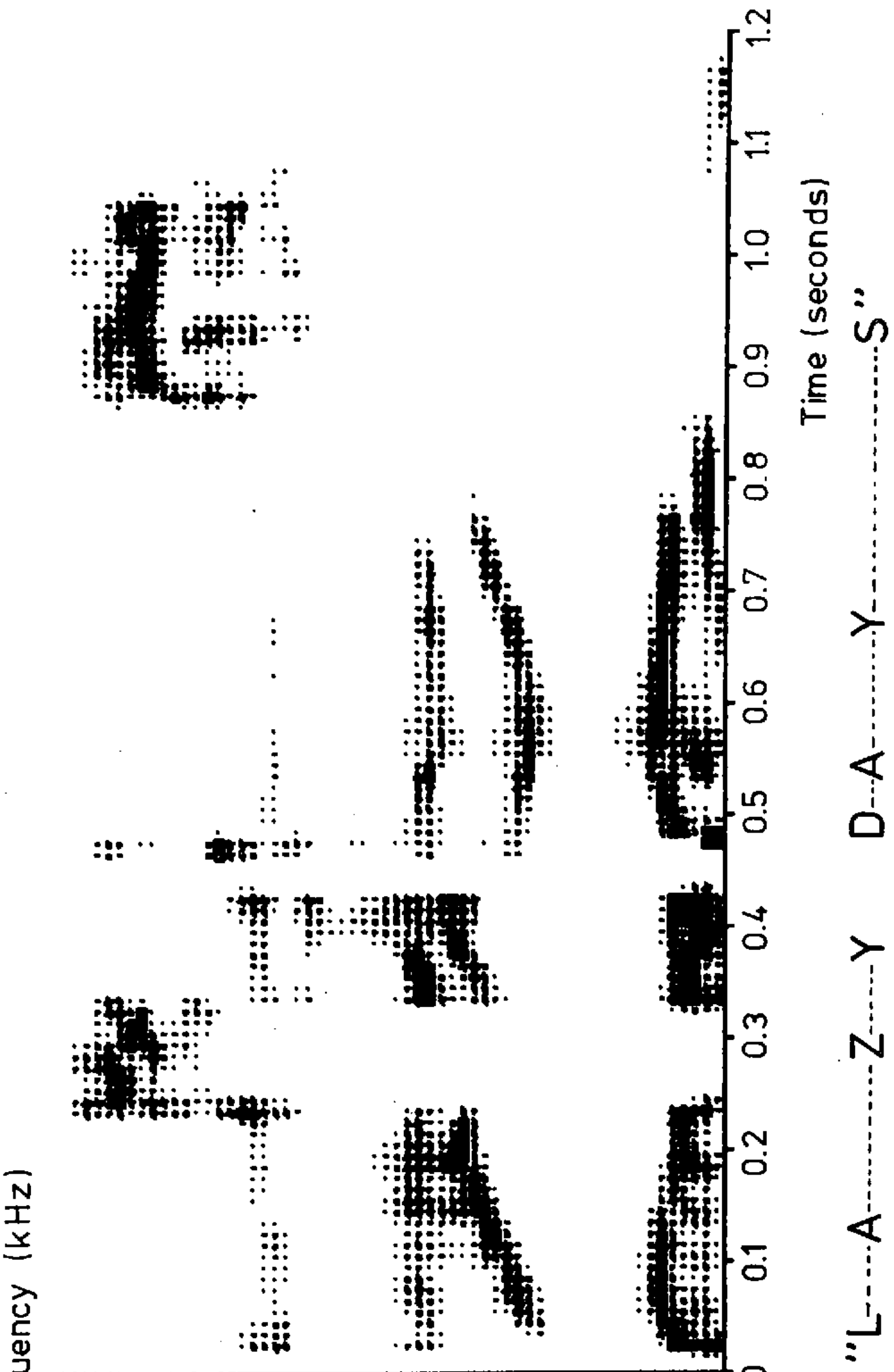
Frequency (kHz)
6.0
5.5
5.0
4.5
4.0
3.5
3.0
2.5
2.0
1.5
1.0
0.5
0
0.1
0.2
0.3
0.4
0.5
0.6
0.7
0.8
0.9
1.0
1.1
1.2
Time (seconds)
"L---A---Z---Y D-A---Y---S"

Figure 3

FFT-based Spectrum Analyser". Electronic Design, 19/8/82, p 149.

[8] C.N. JUDICE, R.D.SLUSKY (1981) "Processing Images for Bilevel Digital Displays". From "Advances in Image Pickup and Display", Volume 4. Academic Press.

APPENDIX - DERIVATION OF THE 5 POINT WINOGRAD SMALL DFT

The 5 point DFT can be represented by the following matrix operation:

$$\begin{bmatrix} X_0 \\ X_1 \\ X_2 \\ X_3 \\ X_4 \end{bmatrix} = \begin{bmatrix} 1 & 1 & 1 & 1 & 1 \\ 1 & w^1 & w^2 & w^3 & w^4 \\ 1 & w^2 & w^4 & w^1 & w^3 \\ 1 & w^3 & w^1 & w^4 & w^2 \\ 1 & w^4 & w^3 & w^2 & w^1 \end{bmatrix} \begin{bmatrix} x_0 \\ x_1 \\ x_2 \\ x_3 \\ x_4 \end{bmatrix}$$

where $X_n = X[n]$

$x_n = x[n]$

$w^n = w[n] = e^{-j2\pi n/5}$

Reordering the matrix according to the Rader prime algorithm, by exchanging the middle two columns, exchanging the last two columns, then exchanging the last two rows produces the following operation:

$$\begin{bmatrix} X_0 \\ X_1 \\ X_2 \\ X_4 \\ X_3 \end{bmatrix} = \begin{bmatrix} 1 & 1 & 1 & 1 & 1 \\ 1 & w^1 & w^3 & w^4 & w^2 \\ 1 & w^2 & w^1 & w^3 & w^4 \\ 1 & w^4 & w^2 & w^1 & w^3 \\ 1 & w^3 & w^4 & w^2 & w^1 \end{bmatrix} \begin{bmatrix} x_0 \\ x_1 \\ x_2 \\ x_4 \\ x_3 \end{bmatrix}$$

From this we get

$$X_0 = x_0 + x_1 + x_2 + x_4 + x_3$$

and

$$\begin{bmatrix} X_1 - X_0 \\ X_2 - X_0 \\ X_4 - X_0 \\ X_3 - X_0 \end{bmatrix} = \begin{bmatrix} w^1-1 & w^3-1 & w^4-1 & w^2-1 \\ w^2-1 & w^1-1 & w^3-1 & w^4-1 \\ w^4-1 & w^2-1 & w^1-1 & w^3-1 \\ w^3-1 & w^4-1 & w^2-1 & w^1-1 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_4 \\ x_3 \end{bmatrix}$$

which can be recognised as a cyclic convolution of the form:

$$\begin{bmatrix} y_1 \\ y_2 \\ y_3 \\ y_4 \end{bmatrix} = \begin{bmatrix} h_1 & h_4 & h_3 & h_2 \\ h_2 & h_1 & h_4 & h_3 \\ h_3 & h_2 & h_1 & h_4 \\ h_4 & h_3 & h_2 & h_1 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \end{bmatrix}$$

AUDIO SPECTRAL ANALYSIS USING THE WINOGRAD FOURIER TRANSFORM

Using the Winograd algorithm for cyclic convolution, the convolution y[n] = h[n] x x[n] can be calculated by the following matrix operation:

$$\begin{bmatrix} y_1 \\ y_2 \\ y_3 \\ y_4 \end{bmatrix} = \begin{bmatrix} 1 & 1 & 1 & 0 & -1 \\ 1 & -1 & 1 & 1 & 0 \\ 1 & 1 & -1 & 0 & 1 \\ 1 & -1 & -1 & -1 & 0 \end{bmatrix} \begin{bmatrix} H_1 & 0 & 0 & 0 & 0 \\ 0 & H_2 & 0 & 0 & 0 \\ 0 & 0 & H_3 & 0 & 0 \\ 0 & 0 & 0 & H_4 & 0 \\ 0 & 0 & 0 & 0 & H_5 \end{bmatrix} \begin{bmatrix} 1 & 1 & 1 & 1 \\ 1 & -1 & 1 & -1 \\ 1 & 1 & -1 & 1- \\ 1 & 0 & -1 & 0 \\ 0 & 1 & 0 & -1 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \end{bmatrix}$$

where

$$\begin{bmatrix} H_1 \\ H_2 \\ H_3 \\ H_4 \\ H_5 \end{bmatrix} = \frac{1}{4} \begin{bmatrix} 1 & 1 & 1 & 1 \\ 1 & -1 & 1 & -1 \\ 2 & 0 & -2 & 0 \\ -2 & 2 & 2 & -2 \\ 2 & 2 & -2 & -2 \end{bmatrix} \begin{bmatrix} h_1 \\ h_2 \\ h_3 \\ h_4 \end{bmatrix}$$

This gives

$$\begin{bmatrix} X_1-X_0 \\ X_2-X_0 \\ X_4-X_0 \\ X_3-X_0 \end{bmatrix} = \begin{bmatrix} 1 & 1 & 1 & 0 & -1 \\ 1 & -1 & 1 & 1 & 0 \\ 1 & 1 & -1 & 0 & 1 \\ 1 & -1 & -1 & -1 & 0 \end{bmatrix} \begin{bmatrix} M_1 & 0 & 0 & 0 & 0 \\ 0 & M_2 & 0 & 0 & 0 \\ 0 & 0 & M_3 & 0 & 0 \\ 0 & 0 & 0 & M_4 & 0 \\ 0 & 0 & 0 & 0 & M_5 \end{bmatrix} \begin{bmatrix} 1 & 1 & 1 & 1 \\ 1 & -1 & 1 & -1 \\ 1 & 1 & -1 & -1 \\ 1 & 0 & -1 & 0 \\ 0 & 1 & 0 & -1 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_4 \\ x_3 \end{bmatrix}$$

Reversing the first two stages of this derivation allows the complete transform to be represented in matrix form as follows:

$$\begin{bmatrix} X_0 \\ X_1 \\ X_2 \\ X_3 \\ X_4 \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 \\ 1 & 1 & 1 & 1 & 0 & -1 \\ 1 & 1 & -1 & 1 & 1 & 0 \\ 1 & 1 & -1 & -1 & -1 & 0 \\ 1 & 1 & 1 & -1 & 0 & 1 \end{bmatrix} \begin{bmatrix} M_0 & 0 & 0 & 0 & 0 & 0 \\ 0 & M_1 & 0 & 0 & 0 & 0 \\ 0 & 0 & M_2 & 0 & 0 & 0 \\ 0 & 0 & 0 & M_3 & 0 & 0 \\ 0 & 0 & 0 & 0 & M_4 & 0 \\ 0 & 0 & 0 & 0 & 0 & M_5 \end{bmatrix} \begin{bmatrix} 1 & 1 & 1 & 1 & 1 \\ 0 & 1 & 1 & 1 & 1 \\ 0 & 1 & -1 & -1 & 1 \\ 0 & 1 & -1 & 1 & -1 \\ 0 & 1 & 0 & 0 & -1 \\ 0 & 0 & -1 & 1 & 0 \end{bmatrix} \begin{bmatrix} x_0 \\ x_1 \\ x_2 \\ x_3 \\ x_4 \end{bmatrix}$$

where

$$M_0 = 1$$

and

AUDIO SPECTRAL ANALYSIS USING THE WINOGRAD FOURIER TRANSFORM

$$\begin{bmatrix} M_1 \\ M_2 \\ M_3 \\ M_4 \\ M_5 \end{bmatrix} = \frac{1}{4} \begin{bmatrix} 1 & 1 & 1 & 1 \\ 1 & -1 & 1 & -1 \\ 2 & 0 & -2 & 0 \\ -2 & 2 & 2 & -2 \\ 2 & 2 & -2 & -2 \end{bmatrix} \begin{bmatrix} w^1 - 1 \\ w^2 - 1 \\ w^4 - 1 \\ w^3 - 1 \end{bmatrix}$$

Institute of Acoustics